

NASA Technical Memorandum 100499

LANGLEY VEDIT FOR NOS/VE  
USAGE MANUAL

M. A. HEANEY

SEPTEMBER 1987

{NASA-TM-100499} LANGLEY VEDIT FOR NOS/VE  
USAGE MANUAL {NASA} 40 p Avail: NTIS HC  
A03/MF A01 CSCL 09B

N87-30098

Unclas  
G3/61 0103541

**NASA**  
National Aeronautics and  
Space Administration  
**Langley Research Center**  
Hampton, Virginia 23665

RELATED MANUALS

XEDIT Version 3 Reference Manual. Control Data Corporation  
Publication Number 60455730, 1984.

Langley XEDIT Reference Manual. Central Scientific Computer  
Complex Documentation N-7a, 1984.

SCL Language Definition/Usage. Control Data Corporation  
Publication Number 60464013, 1986.

SCL System Interface/Usage. Control Data Corporation Publication  
Number 60464014, 1986.

SCL Quick Reference. Control Data Corporation Publication Number  
60464018, 1986.

SCL Advanced File Management/Usage. Control Data Corporation  
Publication Number 60486413, 1986.

## CONTENTS

INTRODUCTION . . . . .	1
AUDIENCE . . . . .	1
VEDIT FEATURES . . . . .	1
REFERENCE MANUALS CONVENTIONS . . . . .	1
CALLING VEDIT . . . . .	3
INITIATING VEDIT . . . . .	3
INTERACTIVE USAGE OF VEDIT . . . . .	3
VEDIT CONVENTIONS . . . . .	4
VEDIT COMMAND SYNTAX . . . . .	4
ENTERING EDITING DATA . . . . .	4
POSITIONING THE FILE POINTER . . . . .	5
LOCATING LINES VIA SPECIFIED STRINGS . . . . .	5
ADVANCING AND REVERSING THE POINTER . . . . .	5
POSITIONING POINTER AT TOP AND BOTTOM OF THE FILE . . . . .	5
POSITIONING POINTER BY LISTING LINES . . . . .	6
STRING EDITING . . . . .	7
REPLACING, DELETING, AND INSERTING STRINGS . . . . .	7
LINE EDITING . . . . .	8
TERMINATING VEDIT . . . . .	9
TERMINATING VEDIT WITH EDITING CHANGES SAVED . . . . .	9
TERMINATING VEDIT WITHOUT SAVING EDITING CHANGES . . . . .	9
COMMANDS AND SUBCOMMANDS . . . . .	11
COMMAND AND SUBCOMMAND FORMAT . . . . .	13
COMMAND AND SUBCOMMAND DESCRIPTION FORMAT . . . . .	14
CHANGE_STRING . . . . .	15
CHANGES_STRING . . . . .	17
DELETE_LINE . . . . .	18
EDIT_FILE_LINE_MODE . . . . .	19
END . . . . .	20
INSERT_LINE . . . . .	22
INSERT_LINE_BEFORE . . . . .	24
LOCATE_STRING . . . . .	26
MOVE_TO_BOTTOM . . . . .	27
MOVE_TO_NEXT_LINE . . . . .	28
MOVE_TO_TOP . . . . .	29
PRINT_LINE . . . . .	30
STOP . . . . .	31
GLOSSARY . . . . .	33
SUMMARY OF VEDIT COMMANDS AND SYNTAX . . . . .	39

## INTRODUCTION

This manual contains information concerning the Langley VEDIT utility line editor on the Control Data Network Operating System/Virtual Environment (NOS/VE).

The EDIT\_FILE\_LINE\_MODE (or VEDIT) command utility enables you to edit files (local or permanent) line by line. These files can contain either source programs, data, or text (listable files). The user should not use this utility to edit binary files.

This manual explains how to access and use VEDIT. Examples and illustrations help the user with each command.

### AUDIENCE

This manual is intended for a user which is familiar with the interactive use of NOS/VE. Some knowledge of how to manipulate NOS/VE files is helpful but is not required. For information on the interactive use of NOS/VE, refer to the System Command Language (SCL) System Interface Manual.

This manual is divided into two parts. The first part describes the basics of the editor. The second part describes all the utility subcommands. This information is listed alphabetically by subcommand names.

### VEDIT FEATURES

The following are features of VEDIT:

- Simple command formats using the same syntax as SCL commands.
- Verification of user entries; VEDIT automatically lists the file lines modified by a command.
- Easy line modification.
- Available for batch processing.

### REFERENCE MANUALS CONVENTIONS

The EDIT\_FILE\_LINE\_MODE command utility is an extended interactive, line-oriented text editor developed by Margaret A. Heaney, CSB/ACD, NASA Langley, based on XEDIT on the Control Data Corporation Network Operating System (CDC NOS) and using command utilities on NOS/VE.

This manual describes this utility's use. The basis of the material for the commands and their syntax has been taken from the CDC XEDIT Version 3 Reference Manual (publication number

60455730). It is assumed that the user is familiar with NOS/VE operations and file concepts.

The subcommand descriptions within this manual follow a standard, concise format. The format is described on page 16 of this manual and is the same format used for other SCL commands (refer to the SCL Language Definition manual for detailed information). In the descriptions of subcommands or parameters, valid abbreviations are shown in parentheses.

In the examples, the user input is printed in lower case letters, and VEDIT output is printed in uppercase letters.

## CALLING VEDIT

The user can call VEDIT from a time-sharing terminal by using the EDIT\_FILE\_LINE\_MODE command, where editing commands can be given interactively. In a batch environment the EDIT\_FILE\_LINE\_MODE utility call implements the utility, where a set of utility sub-commands is processed without further user direction.

### INITIATING VEDIT

The user initiates VEDIT by using the format:

```
EDIT_FILE_LINE_MODE(EDIFLM, VEDIT)
  INPUT_FILE
  FSTAT
  INPUT
  OUTPUT
  STATUS
```

The INPUT\_FILE(F) parameter specifies the name of the file you want to edit and is a required parameter.

If the FSTAT parameter is specified as a key of "c" (at present, the only available option), VEDIT begins execution in creation mode. While in creation mode, only the following commands are valid: INSERT\_LINE and STOP. VEDIT leaves creation mode when a line of text exists within the file. Normal editing can follow. Upon exit, a file is generated with the name specified in the VEDIT command. If the full path is not given, the file is created in the user's current working catalog. If omitted, the file must contain at least one line of text; otherwise, the system prints

```
EMPTY OR FILE NOT FOUND
NAME EDIT FILE?
```

### INTERACTIVE USAGE OF VEDIT

The interactive user issues the command to call the VEDIT program. Output is printed at the terminal using the existing character set mode. After the VEDIT command is entered, the system replies:

```
NOS/VE VEDIT date and time
VED/
```

This message indicates that the VEDIT program is initiated and awaiting commands.

## VEDIT CONVENTIONS

All VEDIT commands follow the same basic pattern in their syntax and parameters and in their required editing data. Knowledge of the basic pattern leads to fewer user errors while editing.

### VEDIT COMMAND SYNTAX

VEDIT prints the prompt "VED/" whenever it expects the user to enter a VEDIT command. The syntax for a VEDIT command is the same as SCL commands.

A legal command conforms to the following conventions:

- The command must be spelled correctly.
- Command parameters must follow the sequence shown in that command's description.
- Command parameters must stay within their maximum and minimum numerical limits.
- The command must be entered at the position where the terminal print element stopped after printing the prompt, "VED/".
- The command must be separated from its parameters by a space or other delimiter and, for string parameters, the string delimiter must also be used.
- Extra spaces may appear between command parameters.
- VEDIT initiates its search for a given line or string at the current pointer position.

If the user does not enter a valid command after the prompt "VED/", VEDIT prints an error message and another prompt.

### ENTERING EDITING DATA

VEDIT commands often require that the user enter data indicating how the file should be modified. When editing data is needed, VEDIT prints a single question mark prompt (?). Editing data is requested for the INSERT\_LINE and the INSERT\_LINE\_BEFORE commands.

If the user enters only a carriage return after the single question mark (?), VEDIT prints the prompt "VED/" indicating that the command was not processed and that the user can enter the next editing command.

## POSITIONING THE FILE POINTER

During editing, VEDIT keeps track of its location within the file by means of a pointer. The pointer is always located at the start of the line and is positioned according to these rules.

<u>Condition</u>	<u>Pointer Position</u>
VEDIT is called.	Beginning-of-information.
A command is processed.	Last line processed by the command (usually the last line printed).
Command processing reaches the end-of-information.	The message -- EOI/TOP -- is printed, further processing of the command stops, and the pointer is repositioned at the beginning of the file.

## LOCATING LINES VIA SPECIFIED STRINGS

The user can advance the pointer to a line containing a specific string of alphanumeric characters by issuing the LOCATE\_STRING command. LOCATE\_STRING allows you to find lines containing a specified string. When the user issues the LOCATE\_STRING command, VEDIT prints the specified number of lines containing the specified string. The pointer is positioned at the last line printed.

## ADVANCING AND REVERSING THE POINTER

When the user wants to advance the pointer toward the end of the file, the MOVE\_TO\_NEXT\_LINE command is entered. When the user wants to reverse the pointer toward the beginning-of-information, MOVE\_TO\_NEXT\_LINE command specifying a negative number of lines is used. Reverse pointer movements using the MOVE\_TO\_NEXT\_LINE command are much slower than forward pointer movements.

## POSITIONING POINTER AT TOP AND BOTTOM OF THE FILE

A user can position the pointer at the beginning-of-information by issuing a MOVE\_TO\_TOP command. The user can move the pointer to the bottom of a file by issuing a MOVE\_TO\_BOTTOM command.

## POSITIONING POINTER BY LISTING LINES

A user can list lines from a file with the PRINT\_LINE command. This command begins its listing at the current pointer position. When the listing is finished, the pointer is positioned at the last line printed. If VEDIT reads an end-of-information mark, printing stops and the pointer is positioned at the beginning of the file.

## STRING EDITING

The user can modify specific strings of alphanumeric characters within a file line by issuing one of the string editing commands, `CHANGE_STRING` or `CHANGES_STRING`.

### REPLACING, DELETING, AND INSERTING STRINGS

Both commands can:

- Replace one string with a different string.
- Delete strings from file lines.
- Insert strings at the beginning of a line.

The user can replace one string with a different string of any length and character content with both commands. However, to use the edited file within `NOS/VE`, the user should not create a file line longer than 80 characters. The search for the first specified string starts at the line designated by the current pointer position.

The commands can also be used to delete a string by omitting the replacement string. This command begins editing at the current line position and continues through the specified number of lines.

A user can insert before the first (leftmost) character of a line using these commands. To do this, the replaced string must be omitted.

The `CHANGE_STRING` command changes every occurrence of the specified string in each of the specified number of lines.

The `CHANGES_STRING` command changes a specific number of occurrences of the string in as many lines as necessary.

## LINE EDITING

A user can delete entire lines from a file by using any of the following criteria.

- A sequence of lines starting at the current pointer position.
- Selective lines on the basis of a specified string.
- A single line with a specified string without moving the file pointer if the string is not found.

These criteria use the `DELETE_LINE` command. The `DELETE_LINE` command leaves the file pointer positioned at the line after the last line deleted.

The user can insert entire new lines into a file, without affecting any existing lines, in two different ways.

- `INSERT_LINE` command inserts a specified number of lines after the current pointer position.
- `INSERT_LINE_BEFORE` command inserts a specified number of lines before the current pointer position.

When the user enters the `INSERT_LINE` command, `VEDIT` prints a single question mark to prompt entry for the first line to be inserted. `VEDIT` continues to print single question mark prompts until the specified number of lines to be inserted has been entered. If the user enters a carriage return immediately after the (?) prompt `VEDIT` stops processing the `INSERT` command and requests the next command. To insert a blank line, the user enters a space before the carriage return. After processing an `INSERT_LINE` command, the file pointer is positioned at the last line inserted.

When the user wants to insert a specific number of lines into a file before the line at the current pointer position, an `INSERT_LINE_BEFORE` command is issued. Entry of the lines to be inserted is the same as for the `INSERT_LINE` command. The first line entered is the first line of the insertion sequence within the file. After processing an `INSERT_LINE_BEFORE` command, the pointer is at the same position it occupied before the command was issued.

## TERMINATING VEDIT

The user issues one of the following commands to end the editing session.

END	Terminates editing and saves the edit file.
STOP	Terminates editing and does not save the edit file.

### TERMINATING VEDIT WITH EDITING CHANGES SAVED

When the user wants to terminate VEDIT execution and keep the edited file, an END command is issued. When the command is entered, the edit file is written onto the file specified on the END command. VEDIT then terminates its execution.

### TERMINATING VEDIT WITHOUT SAVING EDITING CHANGES

The user issues a STOP command to terminate VEDIT without saving the edited version of the file. The file is left exactly as it appeared before VEDIT was called into execution.

## COMMANDS AND SUBCOMMANDS

This section contains individual descriptions of each VEDIT sub-command; it also contains individual descriptions of the command that initiates VEDIT execution. The descriptions are presented in alphabetical order by name.

CHANGE_STRING . . . . .	15
CHANGES_STRING . . . . .	17
DELETE_LINE . . . . .	18
EDIT_FILE_LINE_MODE . . . . .	19
END . . . . .	20
INSERT_LINE . . . . .	22
INSERT_LINE_BEFORE . . . . .	24
LOCATE_STRING . . . . .	26
MOVE_TO_BOTTOM . . . . .	27
MOVE_TO_NEXT_LINE . . . . .	28
MOVE_TO_TOP . . . . .	29
PRINT_LINE . . . . .	30
STOP . . . . .	31

PRECEDING PAGE BLANK NOT FILLED

## COMMAND AND SUBCOMMAND FORMAT

The VEDIT subcommands use the SCL command syntax. For a complete description of the SCL command syntax, refer to the SCL Language Definition manual.

The following is a list of reminders when using SCL command syntax.

- The abbreviation for each command, subcommand, or parameter name follows the name in the individual command, subcommand, or parameter description. The standard abbreviation convention for command and subcommand names is the first three characters of the first word plus the first character of each succeeding word.
- Each subcommand also has an alias that closely resembles a NOS XEDIT command.
- A delimiter character must separate the command or subcommand name from the parameter list, each parameter within the parameter list, and each value specified for a parameter. A valid delimiter character is either a comma or a space.
- Each parameter has a name. You specify a parameter value either by specifying the value after the parameter name and an "=" character, or by specifying the value alone in its position within the parameter list. When you specify parameter values using their parameter names, you can enter the parameter values in any order.
- You can use more than one line to enter a subcommand. The second and subsequent lines are called continuation lines. To continue a subcommand on the next line, enter an ellipsis (..) at the end of the line.
- SCL interprets uppercase and lowercase letters the same within a name. However, SCL distinguishes between uppercase and lowercase within a string. Therefore, you must specify uppercase or lowercase letters as found in text when specifying a string search, as in using the LOCATE\_STRING command.

## COMMAND AND SUBCOMMAND DESCRIPTION FORMAT

Each command or subcommand description in this manual provides the following information.

- Command or subcommand name with its abbreviation and aliases.
- Brief statement of the command or subcommand function.
- Format including both singular and plural forms of the command or subcommand name and the position of each parameter in the parameter list.
- Parameter value kinds. Within the command or sub-command format, each parameter name is equated to a word indicating the parameter value kind. For information on parameter value kinds, refer to the SCL Language Definition manual.
- Parameter name forms. Each parameter description lists the singular and plural forms of the parameter name and its abbreviation in parentheses.
- Brief statement of the function of each parameter.
- Indication of whether the parameter is required or optional and, if optional, the default value.
- Additional remarks on command or subcommand options and processing details.
- Brief example using the command or subcommand.

## CHANGE\_STRING

**Purpose** Replaces one string with a different string of any length and character content. However to use the edited file within NOS/VE, the user should not create a file line longer than 80 characters. The search for the first specified string starts at the line designated by the current pointer position.

**Format** CHANGE\_STRING  
or  
CHAS  
or  
CHANGE  
or  
C  
STRING1=string  
STRING2=string  
NUMBER=integer

### Parameters

STRING1 or S1  
String to be replaced. If omitted, STRING2 is inserted before the leftmost character of the line(s).

STRING2 or S2  
String to replace string1 (can be any length). If omitted, STRING1 is deleted.

NUMBER or N  
Number of lines containing an occurrence of string1 which should be changed. VEDIT changes every occurrence of string1 in the specified lines. If n is omitted, VEDIT changes the first line found containing string1. Maximum value is 99999. If n is 0, the search for string1 is restricted to the current line.

**Examples** This VEDIT session uses the CHANGE\_STRING command to replace every instance of "EXT " in file to "865-". VEDIT verifies which lines are changed. Then LOCATE\_STRING locates the name "BEE" and changes it to "BEAN"

```
/edit_file_line_mode f=employ_addr
NOS/VE VEDIT January 1, 1987
VED/print 99
##NAMES/ADDRESSES ARE FICTITIOUS##
M. T. JONES
1244 WILSHIRE ST ZIP 55722
EXT 6533
Q. E. SMITH
```

POB 55 ZIP 55703  
EXT 8837  
P. T. BEE  
8710 14TH ST ZIP 55713  
EXT 1890  
-- EOI/TOP --  
VED/change 'ext' '865-' 99  
865-6533  
865-8837  
865-1890  
-- EOI/TOP --  
VED/locate 'bee'  
P. T. BEE  
VED/change 'bee' 'bean'  
P. T. BEAN  
VED/

## CHANGES\_STRING

**Purpose** Replaces one string with a different string of any length and character content. However to use the edited file within NOS/VE, the user should not create a file line longer than 80 characters. The search for the first specified string starts at the line designated by the current pointer position.

**Format** CHANGES\_STRING  
or  
CHANGES  
or  
CS  
STRING1=string  
STRING2=string  
NUMBER=integer

### Parameters

STRING1 or S1  
String to be replaced. If omitted, STRING2 is inserted before the leftmost character of the line(s).

STRING2 or S2  
String to replace string1 (can be any length). If omitted, STRING1 is deleted.

NUMBER or M  
Number of occurrences of string1 that are to be replaced. If the user omits the "m" value, VEDIT replaces only the next occurrence of the string.

**Example** This VEDIT session uses the CHANGES\_STRING command to delete the first occurrence of the string "EE" from a line without changing any other occurrence of the string on the line.

```
VED/print 5
POB 55 ZIP 55703
EXT 8837
P. TEE BEE
8710 14TH ST ZIP 55713
EXT 1890
-- EOI/TOP --
VED/changes 'ee' '.'
P. T. BEE
VED/
```

## DELETE\_LINE

**Purpose** Removes an entire line(s); leaves the file pointer positioned at the line after the last line deleted.

**Format** DELETE\_LINE  
or  
DELL  
or  
DELETE  
or  
D  
STRING=string  
NUMBER=integer

### Parameters

STRING or S  
String of characters that determines if a line should be deleted. If omitted, the next "number" lines are deleted.

NUMBER or N  
Number of lines starting at the current pointer position that are to be deleted. The maximum value is 99999; the default value is 1. If n is 0, the line at the current pointer position is deleted.

**Example** This VEDIT session uses the DELETE\_LINE command to delete selective consecutive lines and lines containing a specific string.

```
VED/print
B. P. PEPPERS
VED/delete 3
B. P. PEPPERS
116 WEST ELM DRIVE ZIP 55648
EXT 3222
VED/print
P. T. BEE
VED/delete n=99 s='ext'
EXT 6533
EXT 5339
EXT 67
EXT 1101
-- EOI/TOP --
VED/
```

## EDIT\_FILE\_LINE\_MODE

**Purpose** Begins an Edit\_File\_Line\_Mode\_Utility session. The utility edits the specified file.

**Format** EDIT\_FILE\_LINE\_MODE  
          or  
          EDIFLM  
          or  
          VEDIT  
          INPUT\_FILE= file  
          FSTAT= key  
          INPUT= file  
          OUTPUT= file  
          STATUS=status variable

### Parameters

INPUT\_FILE or F  
File to be edited. This is a required parameter.

FSTAT  
KEY = c, if user wishes to create file with VEDIT.

INPUT or I  
File from which VEDIT reads its input. The default value is \$local.command, which is usually connected to the terminal screen.

OUTPUT or O  
File to which VEDIT output is written. The default value is \$OUTPUT, which is the terminal screen.

STATUS or S  
Specifies the termination condition of the command.

**Remarks** The utility session ends when you enter a QUIT, END, or STOP subcommand.

END

**Purpose** Terminates VEDIT execution and keeps the edited file. When the command is entered, the edit file is written onto the specified file (the fname parameter). VEDIT then terminates its execution.

**Format** END  
or  
QUIT  
FNAME=file

**Parameters**

FNAME or F  
File name to be given the edit file upon leaving VEDIT. If the name is not entered, VEDIT uses the name specified for the edit file when VEDIT was called.

**Example** This VEDIT session uses the END command to save the edited file in a new file, leaving the original file unchanged.

```
VED/change 'ext' 'extension' 99
EXTENSION 6533
EXTENSION 5339
EXTENSION 6778
EXTENSION 1101
--EOI/TOP --
VED/end f=address
Changes to file saved in file: .VE1.VERPG.EMPLOYEE_FILES.ADDRESS
/copy_file employ_addr
M. T. JONES
1544 WILSHIRE ST ZIP 55722
EXT 6533
A. B. MACDONALD
1313 LEMONTREE AVE ZIP 55722
EXT 5339
T. G. SLATER
322 WILSHIRE ST ZIP 55722
EXT 6778
R. C. CARTER
6100 WILSHIRE ST ZIP 55722
EXT 1101
/copy_file address
M. T. JONES
1544 WILSHIRE ST ZIP 55722
EXTENSION 6533
A. B. MACDONALD
1313 LEMONTREE AVE ZIP 55722
EXTENSION 5339
```

T. G. SLATER  
322 WILSHIRE ST ZIP 55722  
EXTENSION 6778  
R. C. CARTER  
6100 WILSHIRE ST ZIP 55722  
EXTENSION 1101

## INSERT\_LINE

**Purpose** Insert a specific number of lines into a file immediately after the line at the current pointer position.

**Format** INSERT\_LINE  
or  
INSL  
or  
INSERT  
or  
I  
NUMBER=integer

### Parameters

NUMBER or N  
Number of lines which the user wants to insert. The maximum value is 99999; the default value is 1. If n is 0, VEDIT assumes that one line is to be inserted.

**Remarks** When the user enters the INSERT command, VEDIT prints a single question mark to prompt entry for the first line to be inserted. VEDIT continues to print single question mark prompts until the specified number of lines to be inserted has been entered. If the user enters a carriage return immediately after the ? prompt, VEDIT stops processing the INSERT command and requests the next command. To insert a blank line, the user enters a space before the carriage return. After processing an INSERT command, the file pointer is positioned at the last line inserted.

**Example** This VEDIT session uses the INSERT\_LINE command to add three lines.

```
VED/print
EXT 6533
VED/insert 3
? a. b. newton
? 160 haskell circle zip 55713
? ext 227
VED/top
VED/print 7
M. T. JONES
1544 WILSHIRE ST ZIP 55713
EXT 6533
A. B. NEWTON
160 HASKELL CIRCLE ZIP 55713
EXT 227
```

T. T. SLATER  
VED/

## INSERT\_LINE\_BEFORE

**Purpose** Insert a specific number of lines into a file immediately before the line at the current pointer position.

**Format** INSERT\_LINE\_BEFORE  
or  
INSLB  
or  
INSERTB  
or  
IB  
NUMBER=integer

### Parameters

NUMBER or N  
Number of lines which the user wants to insert. The maximum value is 99999; the default value is 1. If n is 0, VEDIT assumes that one line is to be inserted.

**Remarks** When the user enters the INSERTB command, VEDIT prints a single question mark to prompt entry for the first line to be inserted. VEDIT continues to print single question mark prompts until the specified number of lines to be inserted has been entered. If the user enters a carriage return immediately after the ? prompt, VEDIT stops processing the INSERTB command and requests the next command. To insert a blank line, the user enters a space before the carriage return. After processing an INSERTB command, the file pointer is at the position it occupied before the command was issued.

**Example** This VEDIT session uses the INSERT\_LINE\_BEFORE command to insert three lines before the pointer position.

```
VED/ print
T. G. SLATER
VED/insertb 3
? a. p. jones
? 160 hood ave zip 55713
? ext 0842
VED/top
VED/print 7
M. T. JONES
1544 WILSHIRE ST ZIP 55713
EXT 6533
A. P. JONES
160 HOOD AVE ZIP 55713
EXT 0842
```

T. G. SLATER  
VED/

## LOCATE\_STRING

**Purpose** Allows the user to advance the pointer to a line containing a specific string of alphanumeric characters.

**Format** LOCATE\_STRING  
or  
LOCS  
or  
LOCATE  
or  
L  
STRING=string  
NUMBER=integer

### Parameters

STRING or S  
String of alphanumeric characters which VEDIT attempts to locate.

NUMBER or N  
Number of lines to be located containing the specified string. The pointer is positioned at the last line located. The maximum value is 99999; the default value is 1. If n is 0, the search is restricted to the current line.

**Remarks** When the user issues a LOCATE\_LINE command, VEDIT prints the n lines containing the specified string. The pointer is positioned at the last line printed. If the string is not found, the pointer is positioned at the top.

**Example** This VEDIT session uses the LOCATE\_STRING command to find a specified string and demonstrates the error message that is displayed if the specified string is not contained in the file.

```
VED/locate 'zip'  
1544 WILSHIRE ST ZIP 55722  
VED/ locate '8711'  
-- EOI/TOP --  
STRING NOT FOUND  
VED/
```

## MOVE\_TO\_BOTTOM

**Purpose**    Position the pointer at the End-of-Information.

**Format**    MOVE\_TO\_BOTTOM  
              or  
              MOVTB  
              or  
              BOTTOM  
              or  
              B

**Example**    This VEDIT session uses the MOVE\_TO\_BOTTOM command to  
              move the pointer to the last line of the file.

```
VED/bottom
EXT 1890
VED/print 5
EXT 1890
-- EOI/TOP --
VED/
```

## MOVE\_TO\_NEXT\_LINE

**Purpose** Allows the user to advance the pointer to the next line of the file in a forward or reverse direction.

**Format** MOVE\_TO\_NEXT\_LINE  
          or  
          MOVTNL  
          or  
          NEXT  
          or  
          N  
          NUMBER=integer

### Parameters

NUMBER or N  
Number of lines that the pointer is advanced. To move in the reverse direction, a negative value is entered. The maximum value is 99999; the default value is 1.

**Remarks** Reverse pointer movements are much slower than forward pointer movements.

**Example** This VEDIT session uses the MOVE\_TO\_NEXT\_LINE command to move the pointer through the file.

```
VED/print 6
M. T. JONES
1544 WILSHIRE ST ZIP 5572
EXT 6533
A. B. NEWTON
166 HASKELL CIRCLE ZIP 55713
EXT 227
VED/next -3
EXT 6533
VED/
```

## MOVE\_TO\_TOP

**Purpose** Position the pointer at the beginning-of-information

**Format** MOVE\_TO\_TOP  
          or  
          MOVTT  
          or  
          TOP  
          or  
          T

**Example** This VEDIT session uses the MOVE\_TO\_TOP command to move the pointer to the first line of the file.

```
VED/print
Q. E. SMITH
VED/top
VED/print
## NAMES/ADDRESSES ARE FICTITIOUS ##
VED/
```

## PRINT\_LINE

**Purpose** List lines from the current pointer position.

**Format** PRINT\_LINE  
or  
PRIL  
or  
PRINT  
or  
P  
NUMBER=integer

### Parameters

NUMBER or N  
Number of lines the user wants printed. The maximum value is 99999; the default value is 1.

**Remarks** When the listing is finished, the pointer is positioned at the last line printed. If an end-of-information is read, printing stops and the pointer is positioned at the beginning of the file.

**Example** This VEDIT session uses the PRINT\_LINE command to display the file.

```
VED/print
Q. E. SMITH
VED/print 4
Q. E. SMITH
POB 55 ZIP 55713
EXT 8760
-- EOI/TOP --
VED/print 2
## NAMES/ADDRESSES ARE FICTITIOUS ##
M. T. JONES
VED/
```

## STOP

**Purpose** Terminates VEDIT without saving the edited version of the file. The file is left exactly as it appeared before VEDIT was called into execution.

**Format** STOP

**Example** This VEDIT session is ended using the STOP command. The changes made to the file are not saved.

```
/edit_file_line_mode f=address
NOS/VE VEDIT JANUARY 12, 1987
VED/delete 10
R. C. CARTER
6100 WILSHIRE ST ZIP CODE 55722
EXT 1101
-- EOI/TOP --
VED/stop
CHANGES MADE TO FILE NOT SAVED.
/edit_file_line_mode f=address
NOS/VE VEDIT JANUARY 12, 1987
VED/print 10
R. C. CARTER
6100 WILSHIRE ST ZIP CODE 55722
EXT 1101
-- EOI/TOP --
VED/
```

## GLOSSARY

### Alphanumeric

Consisting of alphanumeric and/or numeric characters only.

### Batch Mode

A mode of execution where a job is submitted and processed as a unit without intervention from the user. Contrast with Interactive Mode.

### Beginning-of-Information (BOI)

File boundary that marks the beginning of a file.

### Binary File

A noneditable file containing non-text/non-coded format.

### BOI

See Beginning-of-Information.

### Catalog

A directory of files and catalogs maintained by the system for a user. The catalog \$LOCAL contains only file entries.

Also the part of a path that identifies a particular catalog in a catalog hierarchy. The format is as follows:

name.name. . . .name

where each name is a catalog. See Catalog Name and Path.

### Catalog Name

The name of a catalog in a catalog hierarchy (path). By convention, the name of the user's master catalog is the same as the user's user name.

### Command

A set of characters used to signify a specific set of instructions in a time-sharing subsystem. Commands that can only be entered from within the editor are editor subcommands.

### Command List

One or more entries that define the commands that are currently available. A command list entry is an object library, catalog, or special entry \$SYSTEM.

### Command Utility

A NOS/VE processor that adds its command table (referred to as its subcommands) to the beginning of the SCL command list. The subcommands are removed from the command list when the processor terminates.

### Control Statement

A set of characters used to signify a specific set of instructions for a batch job.

### Current Line

Line at which the pointer is positioned. The last line processed by a command (usually the last line printed).

### Default Value

A predetermined value assigned to a parameter.

### Delimiter

The string delimiter is a single quote ('). Parameter delimiters can be any valid NOS/VE delimiter.

### End-of-Information (EOI)

The point at which data in the file ends. VEDIT prints the -- EOI/TOP -- message when EOI is encountered.

-- EOI/TOP --

A message printed when command processing reaches the end of the file being edited.

### Family

A logical grouping of NOS/VE users that determines the location of their permanent files. A family can be subdivided into accounts and projects.

### Family Name

A name that identifies a NOS/VE family. See Family.

#### Family Path

Identifies a file via a family name and a user path as follows:

:family name.user path  
OR  
\$FAMILY.user path

#### File

An SCL element specifying a temporary or permanent file, including its path and, optionally, a cycle reference (for permanent files). A file is identified by specifying a path and, optionally, a cycle reference as follows:

path.cycle reference

A set of information that begins at beginning-of-information (BOI) and ends at end-of-information (EOI), and is referred to by a file name.

#### File Name

The name of a NOS/VE file. It is used in a file reference to identify the file. See Name.

#### Interactive Mode

A mode of execution where a user enters commands at a terminal and each command elicits a response from the computer. Contrast with Batch Mode.

#### Job

A set of tasks executed for a user name. NOS/VE accepts interactive and batch jobs. In interactive mode, a job is usually the same as a terminal session.

#### Local File

A file that is accessed via the local catalog named \$LOCAL. This file no longer exists once the user logs off. See also File, Path, Local Path.

#### Local File Name

The name used by an executing job to reference a file while the file is assigned to the job's \$LOCAL catalog. Only one file can be associated with a given name in one job; however, in one job a file can have more than one instance

of the file open by that name.

#### Local Path

Identifies a local file as follows:

\$LOCAL.file name

#### Path

Identifies a file. It may include a family name, user name, subcatalog name or names, and file name.

#### Permanent Catalog

A catalog of permanent files.

#### Permanent File

A file that does not go away when you log off or when the system is deadstarted. A permanent file has an entry in a permanent catalog. See File.

#### Pointer

A position indicator; in VEDIT, for example, it identifies the line at which VEDIT will start processing a command.

#### SCL

Source Code Language provides the user interface to NOS/VE.

#### Source Program

A program written in a high-level language such as BASIC or FORTRAN and stored in a format that can be edited.

#### String

A set of characters bounded on each end by a single quote mark.

#### Subcommand

A command that can only be entered and recognized from within the editor utility.

#### User Path

Identifies a file or catalog via a user name and optionally a relative path as follows:

.user name.relative path  
OR  
\$USER.relative path

#### Utility

See Command Utility.

#### Working Catalog

The catalog prefixed to a file reference that begins with a name (that is not a period, colon, or a system name beginning with a dollar sign). The working catalog is the catalog used if no other catalog is specified on a file reference.

## SUMMARY OF VEDIT COMMANDS AND SYNTAX

This section gives a short summary of the VEDIT commands and their NOS aliases.

### SUMMARY OF VEDIT COMMANDS

CHANGE_STRING or CHANGE or C	Replaces all occurrences of a particular string of characters with another string in a specified number of lines.
CHANGES_STRING or CHANGES or CS	Replaces one or more occurrences of a particular string of characters with another string.
DELETE_LINE or DELETE or D	Deletes one or more lines from a file starting at the current pointer position or a particular number of lines on the basis of specified string criteria.
END or QUIT	Saves the edited file and terminates VEDIT execution.
INSERT_LINE or INSERT or I	Inserts a specified number of lines into the file immediately after the line designated by the current pointer position.
INSERT_LINE_BEFORE or INSERTB or IB	Inserts a specified number of lines into the file before the line designated by the current pointer position.
LOCATE_STRING or LOCATE or L	Locates a specified number of lines that contain a particular string or strings of characters. The lines are printed.
MOVE_TO_BOTTOM or BOTTOM or B	Finds the end-of-information of the file.
MOVE_TO_NEXT_LINE or NEXT or N	Moves the file pointer from its current position forward or backward a specified number of lines.
MOVE_TO_TOP or TOP or T	Moves the file pointer to the first line in the user's file.

PRINT\_LINE or  
PRINT or P

Lists a specified number of lines,  
starting at the current pointer  
position.

STOP

Terminates VEDIT execution without  
saving the modified edit file.



# Report Documentation Page

1. Report No. NASA TM-100499		2. Government Accession No.		3. Recipient's Catalog No.	
4. Title and Subtitle Langley VEDIT for NOS/VE Usage Manual			5. Report Date September 1987		
			6. Performing Organization Code		
7. Author(s) Margaret A. Heaney			8. Performing Organization Report No.		
			10. Work Unit No.		
9. Performing Organization Name and Address NASA Langley Research Center Hampton, VA 23665			11. Contract or Grant No.		
			13. Type of Report and Period Covered Technical Memorandum		
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546			14. Sponsoring Agency Code		
			15. Supplementary Notes		
16. Abstract <p>This paper details the use of the VEDIT editor on the CDC Network Operating System/Virtual Environment (NOS/VE). The VEDIT, a utility, allows a user to edit files line by line (line mode). How to access and the use of VEDIT is explained. The parameters and the format of the individual commands are defined. Examples are included.</p>					
17. Key Words (Suggested by Author(s)) Line editor VEDIT Files NOS/VE Line mode			18. Distribution Statement Unclassified--Unlimited STAR Category - 61		
19. Security Classif. (of this report) Unclassified		20. Security Classif. (of this page) Unclassified		21. No. of pages 44	22. Price A03